Avoiding Space Robot Collisions Utilizing the NASA/GSFC Tri-Mode Skin Sensor

Final Technical Report
NASA Grant No. NAG5-1468

S. Mahalingam, F.B. Prinz

Avoiding Space Robot Collisions Utilizing the NASA/GSFC Tri-Mode Skin Sensor

Final Technical Report
NASA Grant No. NAG5-1468
10/15/90-11/14/91
submitted to
NASA/Goddard Space Flight Center
Greenbelt, Maryland 20771

S. Mahalingam
Research Assistant, Department of Mechanical Engineering
Carnegie Mellon University

F.B. Prinz
Principal Investigator
Professor, Department of Mechanical Engineering
Director, Engineering Design Research Center
Carnegie Mellon University

Engineering Design Research Center
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Contents

Abstract

Sensor based robot motion planning research has so far primarily focused on mobile robots. Consider, however, the case of a robot manipulator expected to operate autonomously in a dynamic environment where unexpected collisions can occur with many parts of the robot. Only a sensor based system capable of generating collision free paths would be acceptable in such situations. Recently, work in this area has been reported in literature [Lumelsky87c] in which a deterministic solution for 2DOF systems has been generated. The arm was sensitized with a "skin" of infra-red sensors.

We have proposed a heuristic (potential field based) methodology for redundant robots with large dofs. The key concepts are solving the path planning problem by cooperating global and local planning modules, the use of complete information from the sensors and partial (but appropriate) information from a world model, representation of objects with hyper-ellipsoids in the world model, and the use of variational planning. We intend to sensitize the robot arm with a "skin" of capacitive proximity sensors [Vranish92]. These sensors have been developed at NASA, and are exceptionally suited for the space application.

In the first part of the report, we discuss the development and modeling of the capacitive proximity sensor. In the second part we discuss the motion planning algorithm.

Chapter 1

Introduction

Let us consider a redundant robot manipulator operating in a space station. To operate autonomously, it must be provided capabilities to

- Stop at the presence of an unexpected obstacle. (The obstacle could be an astronaut!)
- Maneuver around the obstacle if possible and continue to perform its task.

Two issues have been addressed towards this end in the course of this research.

1.1 Sensing: The Capaciflector

For planning paths in the presence of unknown obstacles, there is a need for an array of sensors on the robot, which can detect an obstacle approaching any part of the robot structure. We shall refer to this array of sensors wrapped around the robot as the "sensing skin". The sensing skin must be able to function reliably in the extreme environment of space and not disturb or be disturbed by neighboring NASA instruments. It should be simple, compact and be incidental to the robot design.

An approach based on an array of capacitors appeared to be promising in solving both the proximity and tactile models [Vranish90]. However, the system must be able to detect objects (including humans) at ranges in excess of one foot so that the robot can react. To obtain such a range, a capacitive sensor typically must be "stood off" from the grounded robot arm a considerable distance

(approximately one inch). This would disfigure the robot arm, causing it to be bulkier than necessary. It would also make cross-talk between the sensor elements more pronounced and would likely impede the flow of heat from the robot arms to outer space (a serious problem for the Flight Telerobotic Servicer (FTS)).

During the process of this research, a sensor was developed which solves these problems and, in so doing, advances the state-of-the-art in capacitive sensor performance. A single element proof-of-principle sensor has been demonstrated on a robot in the GSFC lab. In this demonstration, the robot routinely detects a human or an aluminum truss element at ranges of one foot. Even tiny objects, such as graphite lead in a pencil have been detected at ranges of five inches.

The thrust of the research conducted under this grant was in the modeling of the electromagnetic fields around the capacitive sensor, which is being used in enhancing the performance of the sensor.

Details of the capacitive sensor and the modeling technique used are described in chapter 2.

1.2 Path Planning

The space station environment will be a structured one. This will lend itself to precise representation on a computer. We will refer to this representation of the robot's environment as the World Model. In planning a path for a robot, checks for collisions are made with the objects in the World Model. This class of planning, with an accurate knowledge of the robot's environment is known as Global Planning.

However, the space station environment would be dynamic to some extent, due to moving objects and new structural elements. This could render the World Model inaccurate. In such situations, the robot is equipped with sensors to detect the presence of obstacles. Using only local information obtained from the sensors, a single move is planned which will keep the robot away from the obstacles and move it closer to the goal. After each move the operation is repeated, until the goal is reached. This class of planning is called Local Planning.

It is reasonable to assume that it will be largely possible to maintain an accurate knowledge of the world, and that unexpected obstacles would appear infrequently. It would therefore be warranted to first plan a path using the global technique, and then if an unexpected obstacle is accosted, to modify the path using the local technique.

A number of global planning algorithms for robot manipulators are available [Brooks83], [Gouzènes84], [Laugier85], [Faverjon86], [Lozano-Peréz87], [Barraquand89], [Campbell92]. In this research we have focused on the local planning part of the problem. The essence of the research problem is, then, to solve the path planning problem for a redundant robot manipulator system working in a dynamic, but mostly static, structured, environment.

With the technique of dividing the path planning task between a global and a local planner, the global planner can work with a coarser description of the world, leaving the local planner to take care of obstacle avoidance at the finer level. This provides another motivation for solving this problem.

Our approach to solve this problem, and implementation results are described in chapter 3.

Chapter 2

Sensing

2.1 The "Capaciflector"

The "Capaciflector" (*Capaci*tive Re*flector*) is a capacitive sensing element backed by a reflector element which is driven by the same voltage as the sensor to reflect all field lines away from the grounded robot arm, thus extending the range of the sensor. This approach is an extension of the technique used in instrumentation systems where a shield or guard is used to eliminate stray capacitance [Webster88].

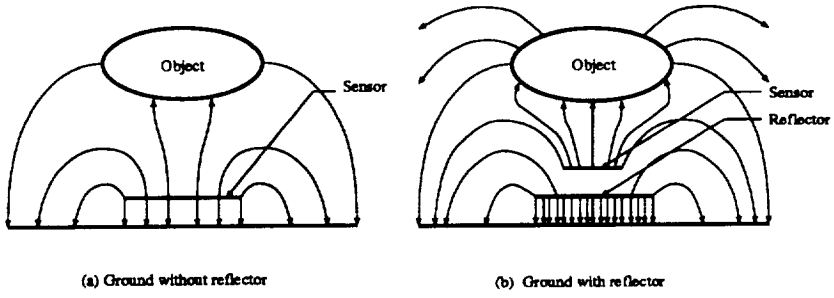

Figure 2.1: "Capaciflector" principle

Fig. 2.1 shows the principles of operation in terms of charges and electric fields. Fig. 2.1a shows a capacitive sensor not using the "capaciflector" principle. Since we are using relatively low frequencies (approximately 20kHz) we have the quasi-static condition and static charges and electric fields can be used to determine

the capacitance the sensor "sees". We can see that the smaller the stand-off from the grounded robot arm, the larger the capacitive coupling between the sensor and the ground. This, of course, has the effect of reducing the relative coupling between the sensor and the object being sensed, and hence reducing sensor range and sensitivity. On the other hand, increasing the stand-off increases the bulk of the robot arm and adds wires and wiring complications. And, when the insulation materials are added to support the stand-off, the ability of the robot arm to dissipate thermal energy into space is reduced. When the "capaciflector" principle is used (Fig. 2.1b), the field lines from the sensor are prevented from returning directly to ground. The effective stand-off is approximately the width of the active shield thickness (on the order of 0.060 inches) and a robot arm with very little bulk, and still have the performance of a large stand-off.

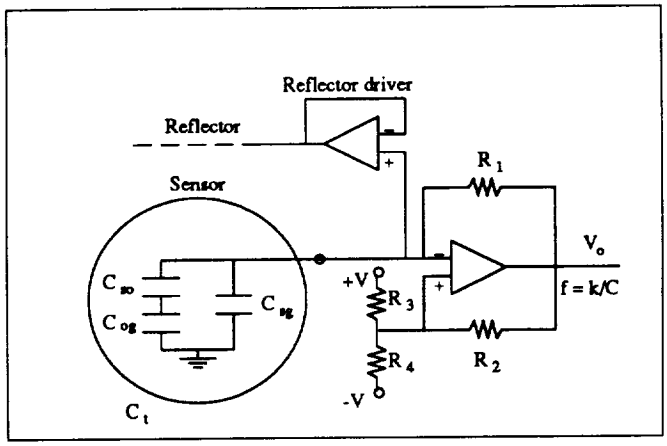

Figure 2.2: "Capaciflector" circuitry

Fig. 2.2 shows the electronic circuitry. The capacitive coupling between the sensor and the object being sensed is used as the input capacitance tuning the oscillator frequency. As an object comes closer, the capacitance increases and the oscillator frequency decreases. On the other hand, the reflector is attached to the output of the voltage follower so it is electrically isolated and prevented from affecting the tuning of the oscillator frequency. At the same time, the voltage of the reflector follows that of the oscillator. Thus, the reflector is in phase with (and reflects) the electric field of the sensor without being affected by the coupling between the sensor and an approaching object.

2.2 Detection

We will now examine the means by which the sensor detects an object. The discussion will be limited to conductors for simplicity although dielectrics are also detected. Both the grounded and ungrounded cases will be examined.

Since we have low frequency, (approximately 20 kHz), the quasi-static case holds. Assuming a momentary positive potential V in Fig. 2.1b, we can see that the electric field lines emanating from the sensor towards the object induce negative charges on the object surface nearest the sensor. Thus that surface can be considered one plate of a capacitor and the senor the other. But, an ungrounded conductive object is charge neutral so an equal amount of positive charge will form on the surface away from the sensor so as to ensure that there is no net electric field in the conductor. These charges couple back to ground which creates a second capacitor in series with the one mentioned above. These are labeled in Fig. 2.2 respectively as C_{so} and C_{og}. But, there also is a path where the electric fields from the sensor can go around the active shield and couple to ground directly. This is labeled as C_{sg}. Thus our tuning capacitance, C_t, is given by the relation

$$C_t = \frac{C_{so}C_{og}}{C_{so} + C_{og}} + C_{sg} \tag{2.1}$$

In the case where the object is grounded, equation 2.1 reduces to

$$C_t = C_{sg} + C_{so} \tag{2.2}$$

Examining equations 2.1 and 2.2 above, since we are looking for small changes in C_t, it is clear we want C_{sg} to be small. Therefore, we want the shield or reflector to force the field lines from the sensor towards the object as much as possible.

We now turn to the case where the object is not grounded [Hayt89, Lorrain88, Fischer89]. We know that

$$C = \frac{Q}{V} \tag{2.3}$$

We also know that a good conductor must have the same potential everywhere on its surface. Therefore the potential on the object will be that of its furthest point from the sensor. We will call the potential on the sensor V and the object potential V_o. Thus we have

$$C_{so} = \frac{Q_i}{V - V_o}; \text{ and} \tag{2.4}$$

$$C_{og} = \frac{Q_i}{V_o} \tag{2.5}$$

where Q_i is the charge induced in the object.

It is apparent that an object with any dimension more than a few inches in any direction (for example length) forces the potential on the entire surface of the object to be very low. And, as the experimental evidence shows, in practice, all objects are approximately grounded.

2.3 Modeling

The frequency of oscillation of the circuit in Fig. 2.2 can be shown to be

$$f = \frac{\ln(0.5)}{2R_1C} \tag{2.6}$$

where $R_3 = R_4 = 2R_2$. This implies

$$\frac{\Delta f}{f_0} = \frac{\Delta C_t}{C_{t0} + \Delta C_t} \tag{2.7}$$

where f_0 and C_{t0} represent the frequency and the capacitance of the sensor in the absence of an object, and Δf and ΔC_t represent the change in frequency and capacitance respectively because of the introduction of an object. Therefore studying the relationship between change in capacitance and sensor configurations is key to improving the sensor's sensitivity. A computer tool was developed towards this end. The tool tracks the change in capacitance as the object moves towards the sensor, and repeats the operations for various configurations.

The modeling is done in a 2D world, the entities comprising the system are assumed to extend to infinity along the axis perpendicular to this plane. Boundary integral method [McAllister85] was used to determine the charge distribution on the entities; the charge distribution trivially leads to determination of the capacitances. The modeling approach is therefore similar to that used by Volakis et al. [Volakis87]. Details of this method are provided in Appendix A.

The system considered in the tool consists of three linear entities (representing the grounded robot arm, the shield, and the sensor), and one circular entity (representing the object). For a given configuration of the ground, shield and sensor, the program tracks the sensor capacitance as the object moves towards the sensor along

the vertical and the horizontal directions. This is done by first defining a planar grid above the sensor, positioning the object center at a grid point, and computing the sensor capacitance for each such object position. The tool output is designed for convenient plotting of the percentage-frequency-change vs distance-from-sensor graphs. The tool automatically generates new configurations (over which the user has control), and performs the above operations for each configuration.

2.4 Results

2.4.1 Experimental and modeling results

An experimental laboratory set-up was assembled. The set-up consisted of a sensor approximately six inches long, the reflector approximately fourteen inches long (Fig. 2.3). The object was one inch in diameter and thirty six inches long. The reflector was made from strips of copper foil that could be connected in the configurations shown in Fig. 2.4. Subsequent testing has shown that the sensor must be shorter than the reflector to reduce the end effects which substantially reduce sensitivity. The explanation is that the reflector must totally surround the sensor to contain the field. Otherwise, the flux lines from the sensor will simply shift to the lower field strength and return to the ground at the ends of the sensor, thereby reducing the coupling to the object. The results of the experiment are shown in Fig. 2.5.

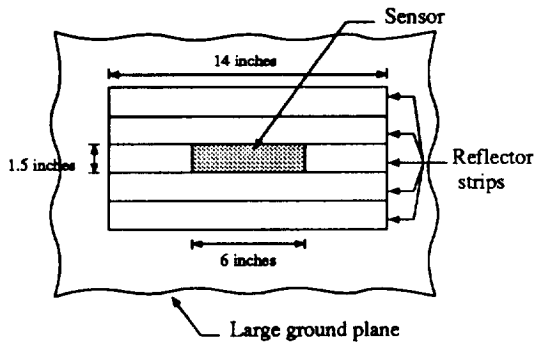

Figure 2.3: Test sensor

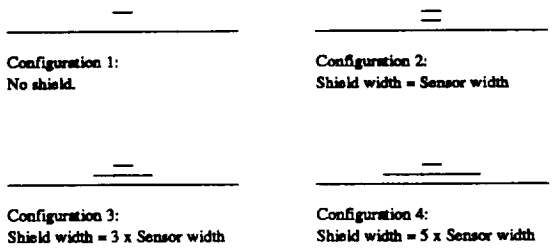

Figure 2.4: Test configurations

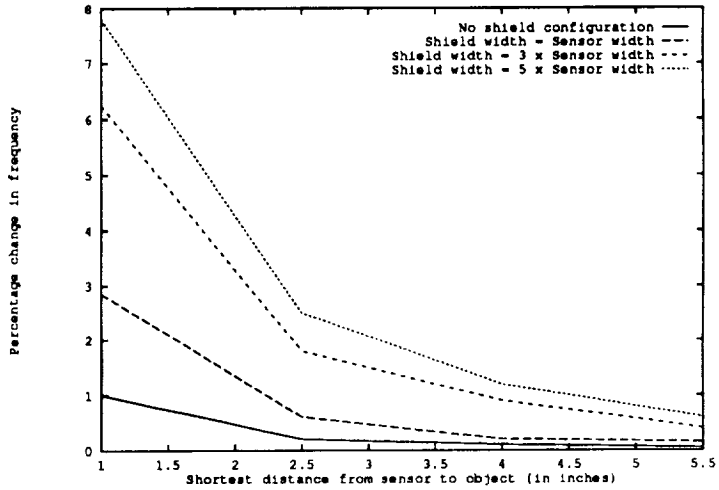

Figure 2.5: Experimental results

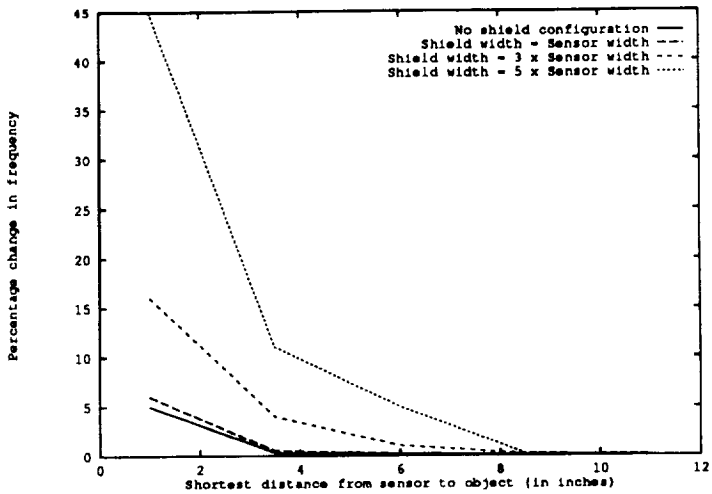

Figure 2.6: Modeling results

The developed tool described in Section 2.3 was used to plot percentage-change-in-frequency vs distance-from-sensor graphs for the four configurations shown in Fig. 2.4. Capacitance is determined using the technique described in Appendix A. Results are shown in Fig. 2.6. As mentioned before, the program computes the sensor capacitance for different object positions. Of these, those positions for which the object's center lies above the center of the sensor are shown in the above graph.

2.4.2 Discussion of results

The results from the modeling and the experiment are similar. Both show the frequency change is inversely proportional to the object distance from the sensor. They both show that the sensitivity increases dramatically as the shield width increases. The increase is approximately seven-fold for the experimental result and almost nine-fold for the model.

The substantial difference shown between the modeled results and the experimental results are probably due to our primitive models used to date. The model program assumes infinitely long strips for the sensor, shield and object, while our experiment used a six inch sensor with fourteen inch shield. End effects or the short sensor may account for the difference; our modeling has not progressed far enough to determine. The rate of variation between the curves is also different. The model shows almost no difference between the curves for no-shield and shield

= sensor width, while the experimental results show a substantial difference. This result may be entirely due to inaccuracies in the model. Similarly, there is a difference between the rate of change between the upper two curves on the graphs. The model shows an increasing rate of change difference while the experimental result shows almost a constant difference. We cannot presently account for this result, but it may be due to either the model or to electronic circuit limitations. This latter conjecture comes from the fact that the frequency changes are substantial and nonlinearities may limit the frequency shift. Investigations are continuing.

Chapter 3

Path Planning

The problem of robot path planning is that of finding a continuous sequence of robot configurations to reach a goal configuration from a given initial configuration. The path should be such that the robot does not collide with any object that may be present in its workspace.

Let us refer to the work space of a robot in R^3 (or R^2, for a planar 2DOF mechanism) as its **Wspace**. The **configuration** ($\bar{q}$) is a specification of the position and orientation of the robot structure with respect to a reference world frame. We parameterize the configuration by $(q_1, q_2, \ldots, q_P)$ for a P-dof system, where each q_i describes the value of a dof. For example, for a robot with rotary joints, each $\bar{q}$ would represent a joint angle value. **Cspace** is the configuration space of the robot. With the parameterization considered, the Cspace would be a P-dimensional Euclidean space.

Fig. 3.1 shows a 2DOF planar robot, the obstacles in its work space (the **W-obstacles**), and the start and goal configurations desired. Fig. 3.2 shows the corresponding Cspace. It should be noted that the robot is mapped to a point in the Cspace. Mappings of the W-obstacles in Cspace are referred to as **C-obstacles**. Therefore the path planning problem in Cspace is one of finding a collision free path for a point automaton amidst C-obstacles. Planning is therefore usually done in Cspace.

Two classes of path planning techniques exist, global and local planning.

Global path planning techniques use *global* information of the robot's world to generate a path. Spatial (and temporal) information of all obstacles within the robot's workspace is presumed to be known. The union of the C-obstacles is called the **C-obstacle region**. The complement of the C-obstacle region is the **free**

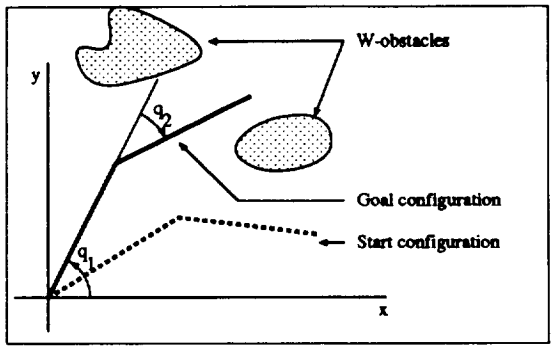

Figure 3.1: Wspace of a 2DOF robot

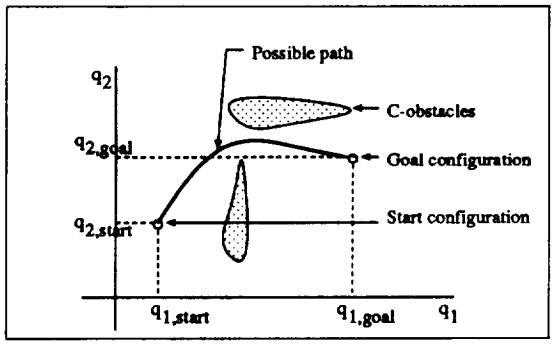

Figure 3.2: Cspace of a 2DOF robot

space. Most techniques capture the connectivity relationship between free spaces in some representation, such as a graph, which is then checked to see whether there is a continuous path between the initial and the goal positions. Such techniques are necessarily done offline due to the computational time required. Therefore, robots operating with such planners will not be able to avoid unexpected obstacles in the planned path.

Local path planning techniques use *local* information of the robot's world for collision avoidance. Sensors are used to retrieve information in the vicinity of the robot, which is used by the planner to avoid collisions. Usually, local planning techniques use the **potential field method**, in which a point robot in its Cspace, moves under the influence of an artificial potential field. The goal forms the attractive potential and the obstacles the repulsive potentials. The robot then moves along the direction of the steepest descent of the potential field. The attraction of this approach is that the planning can be done in real time, no a-priori knowledge of the world is required.

However, since a world model (if one could be available) is not used here, useful, available information is not used. This might result in inefficient, wasted motions. For example, if the world model were queried, it might be clear to the planner that the current path would lead to a dead-end. Instead, the local planner would proceed in the fruitless direction until the sensors detect the dead-end.

These methods can be considered to be two extreme approaches towards obstacle avoidance. Global planners assume that the world is static, and therefore all obstacles are known. Local planners assume that no information of the world is available, therefore all obstacles are unknown.

For a world that is mostly static, such as our prescribed scenario, there is a need for the global and the local planners to interact. The global planner should generate a path with the most current version of the world model. The robot should follow this path until an unexpected object is met with. The local planner should then attempt to direct the robot around the obstacle, and back onto the globally planned path. At this point, the robot should start following the globally planned path.

At a lower level, a robust collision avoidance scheme, designed to continuously monitor the immediate surrounding of the robot, and to keep it away from all objects, is needed to complete the path planning system.

Major local path planning techniques reported in literature are discussed in the sections 3.1.1 and 3.1.2. Discussion is restricted to path planning for robotic manipulators (as opposed to mobile robots). The approach adopted is discussed

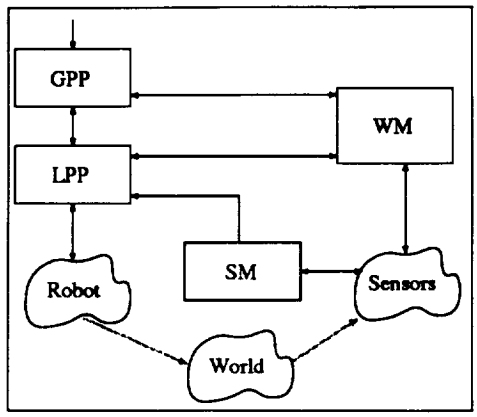

Figure 3.3: Proposed planning architecture

in section 3.2.

3.1 Background

3.1.1 Deterministic methods

While most researchers have utilized heuristic methods for local planning, Lumelsky et al. have extensively explored deterministic methods[1]. Perhaps the greatest contribution of this work is the proof that local feedback information is sufficient to ensure reaching a global objective [Lumelsky87a]. For two DOF robotic systems this approach guarantees to find a solution if one exists, or to correctly conclude that one does not exist. The approach is briefly described below, for a two DOF system.

Let a free path in Cspace be referred as the main (M) line. For every obstacle within the workspace of a robotic manipulator, there exists a shadow region which cannot be accessed by the robot. Let us refer these regions as pseudo obstacles. We refer the mappings of the obstacles and the pseudo obstacles in the robot's configuration space as C-obstacles.

Four classes of M lines have been identified [Lumelsky87c], and a representative M line, a straight line from the goal to the target, is selected as candidate from each class. If an obstacle is detected while following the first selected M

[1] Applicability of these techniques is limited to robotic manipulators.

line (generally the shortest line), the point automaton (mapping of the robot in its Cspace) attempts to follow the contour of the obstacle until it meets the M line behind the obstacle, or until it is determined that the goal cannot be reached by following the current M line. A second M line is then selected, and the operation repeated. If the goal cannot be reached with the second M line, it can be asserted that it is not possible to reach the goal. Contour following is done using the range distance provided by proximity sensors, and knowing the kinematics for the given robot configuration. A priori information about the obstacle is not required.

Although the approach of contour following in Cspace has been extended to a practical three dof robot system [Cheung89b], convergence of the algorithm has not been proved for three and higher dof systems.

3.1.2 Heuristic methods

Heuristic methods *must* be resorted to to plan paths for redundant systems. The most important heuristic method is the potential field approach.

If the obstacles and the robot are similarly charged particles and the goal position has an opposite charge, the robot would be repelled from the obstacles and attracted towards the goal. This is the main concept in the potential field method, proposed by Khatib [Khatib80]. Robot motion is a function of its current position only, and is along the direction of the resultant electrostatic force experienced by the robot at that position. An elemental move is executed along this direction and the forces recalculated, to determine the next motion direction. The process is repeated until the robot reaches the minimum potential. This could either mean that the goal is reached, or the robot is trapped in a local minimum.

The robot links and the obstacles were represented by spheres, capped cylinders, rounded capped cones, and rounded boxes. The point on the robot links closest to an obstacle was used for computing the repulsive force. The distance between the point automaton and its final destination was used to compute the attractive force. The resultant force was defined to be a linear combination of the two forces, and motion was executed along the direction of the resultant force.

The motion generated by this approach is smooth around obstacles. The approach is general, and has been applied to many systems including multiple manipulator coordination [Myers85] and mobile robots [Elfes89].

Major issues in this method are object representation, potential field generation, and negotiating potential wells. The tradeoff in representation is between accuracy and speed of computation. Representing the objects as a combination of

geometric primitives reduces the computation required for determining the potentials; simpler and fewer primitives help this objective. Spheres [2] [Myers85] and ellipsoids [Wang87] have been exclusively used for this purpose, and sometimes with other primitives such as cylinders etc. [Khatib80]. Other representations which have been used are hierarchical bitmaps [Barraquand89], and occupancy grids [Elfes89]. The challenge in generating potential fields is to minimize the number of potential wells and to reduce the depths of the wells [Barraquand89]. Finally, a major drawback of this method has been its inability to efficiently negotiate potential wells. Barraquand has introduced the concept of connectivity graph of local minima to effectively address this issue [Barraquand89]. However, this approach requires global information, and therefore is unsuitable for local path planning applications.

Another drawback of this method is that the potential function to be minimized, is a combination of two opposing functions : the goal or task function which attracts the robot, and the collision avoidance function which repels the robot [Faverjon87]. Depending on the potential function, unsatisfactory results may therefore be produced if an obstacle is very close to the goal.

3.2 Methodology

The large number of DOFs of the system being considered warrants us to use a potential field based approach. We are in the process of implementing an on-line variational approach, which we describe below. Parts of this have been implemented, the results are described in the following section.

It is assumed that a global planner, such as the slide-jump planner [Campbell92], would provide the path to the local path planner (LPP) as a sequence of **path points** in Cspace. We shall refer to this path as the **globally planned path** or the **original path**. Once an obstacle is detected, we attempt to locally modify a portion of the original path, so that the modified path goes around the obstacle. We shall refer to this local modification as **relaxation** of the path (Fig. 3.4). We shall refer to the portion of the path being relaxed as the **path segment** ($\mathcal{S}$) [3]. Let us say that there are R path points in $\mathcal{S}$.

[2] Myers reports that the speed gained by using spheres does not effectively offset the loss in representational accuracy [Myers85].

[3] Work needs to be done to determine what would be a good size for the path segment. This would definitely be dependent on the robot configuration and the geometry of the obstacles.

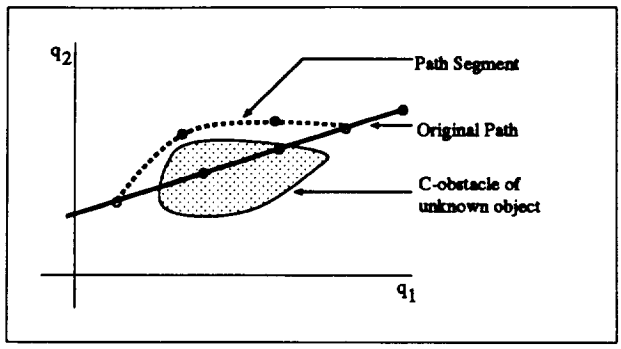

Figure 3.4: Path relaxation

We consider a subset of the Cspace as the search space for the relaxation. We call this subset as the **neighborhood** ($\mathcal{N}$) of $\mathcal{S}$; $\mathcal{S}$ is completely contained within $\mathcal{N}$ (Fig. 3.5). Searching a smaller space for a solution reduces computation time. The dexterity provided by the redundancies enhances the likelihood of finding a collision-free path even in a reduced space[4]. While relaxing $\mathcal{S}$, only those C-obstacles that lie within $\mathcal{N}$ will be considered for planning purposes. This includes all known C-obstacles within that region, which should be obtained by querying a World Model, and the unknown C-obstacles which have been detected by the sensors.

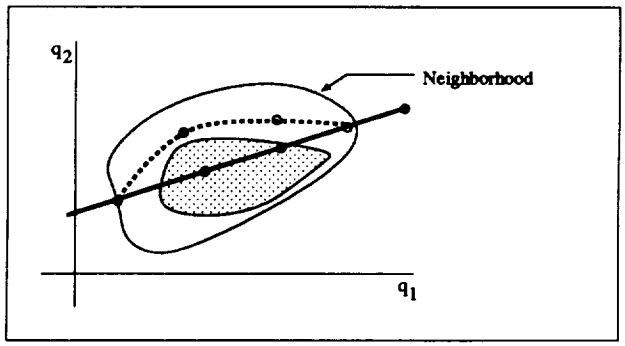

Figure 3.5: Neighborhood of a path segment

We shall use a simplified geometric representation of objects in Wspace (the W-obstacles). We shall use hyper-ellipsoids. We believe these entities will provide

[4]Once again, work needs to be done to determine the size of the neighborhood.

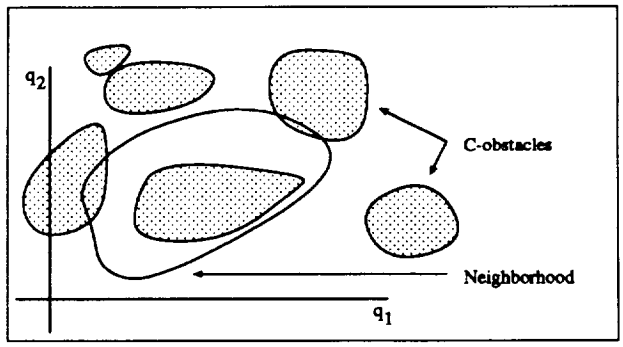

Figure 3.6: C-obstacles

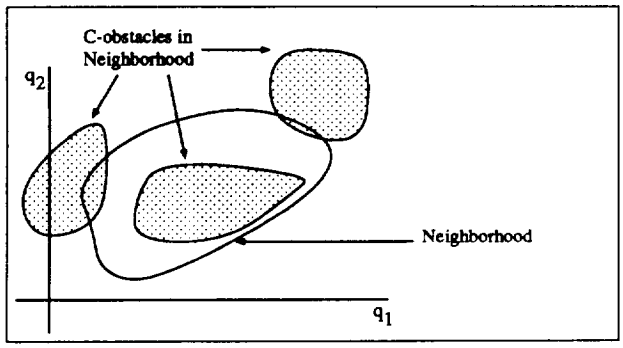

Figure 3.7: C-obstacles in neighborhood

a good tradeoff between accuracy of representation and acceptable computational speed (while computing distances).

We assume that we shall get the hyper-ellipsoid parameters corresponding to each W-obstacle within the mapping of $\mathcal{N}$ in Wspace from the World Model. Corresponding to each sensor which signals the presence of an obstacle, we could represent the unknown W-obstacle by a hyper-ellipsoid of some standard dimensions. Hopefully, the proximity information from the sensors can be used to determine the location of a probable object, such as, say, a girder. Especially, since we do not expect many unknown obstacles to be present. In such a case, we would require lesser number of hyper-ellipsoids for representing the unknown obstacle. Let us say there are N obstacles in $\mathcal{N}$.

As stated in Section 3.1.2, the potential field method is used to maneuver around the newly detected obstacle. Potential fields are produced by the interaction of attractive (robot-goal interaction) and repulsive (robot-obstacle interaction) entities in the Cspace. However, we have the description of the objects in Wspace, and not in Cspace. Generating C-obstacles from W-obstacles is (a) computationally expensive, and (b) non unique for redundant robots. Moreover, we would not have a description of the unexpected obstacle for the mapping. We therefore use the following method, which does not require a description of the C-obstacles, for generating the potential fields in Cspace.

First, let us select some points on the robot structure, which we shall refer to as **control points** (c). These points are used for computing the potential functions, as explained below. Physically, they correspond to the location of the sensors on the robot links. This correspondence is not necessary, but is chosen for convenience. Let us say there are M control points on the robot.

Second, we define the repulsive potential function in the Wspace, the **Wspace potential** (P_{rep}), for ith control point (corresponding to a configuration $\bar{q}$) as

$$P_{i,rep}(c_i(\bar{q})) = \begin{cases} \max_{n=1}^{N}\left(\frac{1}{d(c_i(\bar{q}),O_n)} - \frac{1}{d_0}\right) & \text{if } d(c_i(\bar{q}),O_n) \leq d_0 \\ 0 & \text{if } d(c_i(\bar{q}),O_n) > d_0 \end{cases} \quad \forall i = 1,...,M \tag{3.1}$$

where $d(c_i(\bar{q}), O_n)$ denotes the distance between the ith control point and the nth object, and d_0 the influence distance. If the distance between the control point and the object is greater than d_0, the control point will not be repelled by the object. With this definition, we make P have a large value at the obstacle boundaries.

Finally, we define the potentials in Cspace. The potential of a point $\bar{q}$ in Cspace, the **Cspace potential** (Q), has two components, the repulsive and attractive po-

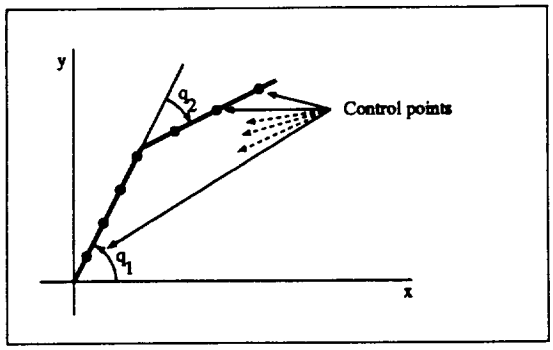

Figure 3.8: Control points

tentials. The repulsive potential is defined as

$$Q_{j,rep}(\bar{q_j}) = \sum_{i=1}^{M} P_{i,rep}(c_i(\bar{q})); \quad \forall j = 1, ..., R \tag{3.2}$$

The attractive potential is defined as

$$Q_{j,att}(\bar{q_j}) = d(\bar{q_j}, \bar{q_{j0}}); \quad \forall j = 1, ..., R \tag{3.3}$$

where $d(\bar{q_j}, \bar{q_{j0}})$ denotes the distance between the jth path point and the position of the same path point in the original path.

Let us now define a **functional** given by

$$J = \int_{\mathcal{S}} (Q_{j,att}(\bar{q_j}) + Q_{j,rep}(\bar{q_j}))d\mathcal{S} \tag{3.4}$$

We define the Cspace potentials such that if $\mathcal{S}$ goes through an object, J would be very large. Our objective is to find a path whose functional would be less than a critical value, $J < J_{cr}$. It is to be noted that we do not attempt to determine the minimum, since our primary aim is to reduce computational time rather than finding an optimal solution.

To determine J, $\mathcal{S}$ is sampled $k(R-1)$ times, where k is an appropriately chosen constant. The effect is to check for collision in between the path points. Another way of achieving the same objective would have been to discretize $\mathcal{S}$ into $k(R-1)$ segments, and sample $\mathcal{S}$ only at the path points. By adopting the former, however, we manage to maintain the dimensionality of the funcional optimization problem to RxP, where P is the number of degree-of-freedoms of the system, in comparison to the kxRxP dimensionality of the latter approach. We use Powell's

N-dimensional minimization algorithm for the minimization. We shall use random walk to escape unacceptable local minima, which in this case would correspond to a path through an object.

The robot is commanded to move towards the first path point in $\mathcal{S}$. The World Model is updated with the sensory information. After some time interval, the local path planner takes another snapshot of the world through the skin sensors, and $\mathcal{S}$ is relaxed again. This process is repeated until the first path point is reached. We shall maintain a constant number of path points in $\mathcal{S}$. Therefore, we delete from $\mathcal{S}$ the path point that has been reached, and include the first path point outside $\mathcal{N}$. The new $\mathcal{N}$ is determined for the modified $\mathcal{S}$. This process is repeated until $\mathcal{S}$ coincides with the original path, at which point the obstacle negotiation would be complete.

Some notes on the control points. As is clear from the above discussion, the variational planning is done in Cspace. Computation of the Cspace potentials would have been far easier if we had a description of the obstacles in Cspace. We do not attempt to map the Wspace obstacles to Cspace due to the complexity of the process involved. Therefore to determine the Cspace potential for a configuration $\bar{q}$, we determine the location of the control points in Wspace corresponding to the configuration, and determine the Wspace potentials for each control point (ref. equation 3.1). Then we compute the Cspace potentials from equation 3.2. Therefore control points are essential for generating the Cspace potentials, which are in turn used in planning.

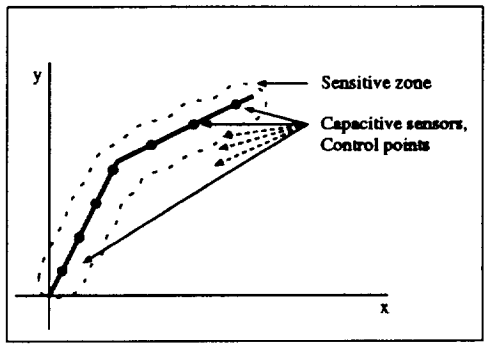

Figure 3.9: Control points and the sensitive zone

By using control points to describe the robot, we essentially discretize the robot for computational purposes. Spacing between the control points is then an

issue. Since collision checks are done only at these points, if they are separated far enough, the planner might not account for some collisions. However, as long as the robot is completely surrounded by the sensitive zone of the capacitive sensors, one can be assured of collision avoidance. For convenience, we have considered the control points to be the sensor locations.

3.3 Current implementation and results

The current version of the LPP has been implemented for a Stanford manipulator. It communicates with a World Model where the objects are represented by ellipsoids. It supports multiple control points. It communicates with a simulation program on the SGI Personal Iris at the Intelligent Robotics Laboratory, NASA/GSFC.

For a three DOF robot, with twelve control points, eight path points in the path segment, and three objects in the neighborhood, it took about 20 seconds to relax a path on a DECStation 3100.

Future work will address the issues of escaping local minima with random walks, representing objects with hyper-ellipsoids, and devicing techniques to further speed up the algorithm.

Appendix A

Charge distribution on the Capaciflector

The given entities are first discretized into M elements. Let an m^{th} element of length Δs_m, be at a potential of V_m, and have a uniform charge distribution of density ρ_m. In our problem, the voltage is specified on all the entities, and hence on all the elements. Let us enable the use of the voltage of any of the elements, say the k^{th} element as the reference voltage. Then applying the boundary condition leads to the following set of $M-1$ equations:

$$\sum_{m=1}^{M} \rho_m K_{nm} = V_n - V_k; \text{ where } n = 1, \ldots, M, \text{ and } n \neq k \qquad \text{(A.1)}$$

where K_{nm} is given by the equation

$$K_{nm} = \frac{1}{2\pi\epsilon_0} \ln \frac{r_{mk}}{r_{nm}}; \text{ where } m \neq n \text{ and } m \neq k \qquad \text{(A.2)}$$

and r_{ij} is the distance between the i^{th} and j^{th} elements. For $m = n$ we have

$$K_{mm} = \frac{1}{2\pi\epsilon_0}(\ln \frac{r_{mk}}{\Delta s_m} + 1 + \ln 2) \qquad \text{(A.3)}$$

and we can show that $K_{nk} = -K_{mm}$. Principle of conservation of charge leads to the last equation:

$$\sum_{m=1}^{M} \Delta s_m \rho_m = 0; \qquad \text{(A.4)}$$

The M linear equations are solved for the charge density distribution on the elements. Capacitance between the sensor and the object is computed from the relation

$$C_t = \sum_{m'=1}^{M'} \Delta s_{m'} \rho_{m'} / (V_{m'} - V_{\text{obj}}) \tag{A.5}$$

where each m' is an element of the sensor, M' is the total number of elements on the sensor, and V_{obj} is the voltage of the object.

Bibliography

[Armstrong89] Armstrong, H.G. *Simulation of vehicle crash and rollover dynamics*, Simulation in Emergency Management and Technology. Proceedings of the SCS Western Multiconference 1989; pp. 13-18; San Diego, CA, USA; 4-6 Jan. 1989

[Barraquand89] Barraquand, J., et al. *Numerical potential field techniques for robot path planning*, Report No. STAN-CS-89-1285, Dept. of Comp. Sci., Stanford Univ., Oct. 1989.

[Brooks83] Brooks, R.A. *Planning collision-free motions for pick-and-place operations*, International Journal of Robotics Research, 2(4).

[Campbell92] Campbell, C.E., Jr. *Fast planning of collision free paths for a three degree of freedom robot*, To appear in Electrical Engineering and Computers: An International Journal, Pergamon Press, 1992.

[Cheung89a] Cheung, E., and Lumelsky, V.J. *Proximity sensing in robot manipulator motion planning: system and implementation issues*, IEEE Trans. on Robotics and Automation, Vol. 5, No. 6, Dec. 1989.

[Cheung89b] Cheung, E., and Lumelsky, V.J. *Development of a sensitive skin for a 3D robot arm operating in an uncertain environment*, Proc. of IEEE Conf. on Robotics and Automation, 1989.

[Elfes89] Elfes, A. *Using occupancy grids for mobile robot perception and navigation*, Computer, Vol.22, No.6, Jun. 1989, pp.46-57.

[Faverjon86] Faverjon, B. *Object level programming of industrial robots*, Proc. of IEEE Conf. on Robotics and Automation, San Fransisco, CA, pp. 1406-1412

[Faverjon87] Faverjon, B., and Tournassoud, P. *A local based approach for path planning of manipulators with a high number of degrees of freedom*, Proc. of IEEE Conf. on Robotics and Automation, Raleigh, NC, 1987, pp. 1152-9.

[Fischer89] Fischer, C.R. *Build the Digital Theramin*, 1989 PE Hobbyists Handbook.

[Gouzènes84] Gouzènes, L. *Strategies for solving collision-free trajetories problems for mobile and manipulator robots*, Internation Journal of Robotics Research, 3(4), 51-65.

[Hayt89] Hayt, Jr. W.H. *Engineering Electromagnetics*, 5th ed. New York: MacGraw-Hill, 1989.

[Khatib80] Khatib, O. Ph.D. Thesis, L'Ecole Nationale Superieure de L'Aeronautique et de L'Espace, Dec. 1980.

[Khatib86] Khatib, O. *Real time obstacle avoidance for manipulators and mobile robots*, Int. J. of Robotics Research, Vol.5, No.1, Spring 1986, pp.90-8.

[Kheradpir88] Kheradpir, S., Thorp, J.S. *Real-time control of robot manipulators in the presence of obstacles*, IEEE J. of Robotics and Automation; vol.4, no.6; Dec. 1988; pp.687-98.

[Laugier85] Laugier, C. and Germain, F. *An adaptive collision-free trajectory planner*, Proc. of the International Conference on Advanced Robotics, Tokyo, Japan, 1985.

[Lorrain88] Lorrain, P., Carson, D.R., and Lorrain, F. *Electromagnetic Fields and Waves*, Freeman, 1988.

[Lozano-Peréz87] Lozano-Peréz, T. *A simple motion planning algorithm for general robot manipulators*, IEEE Transactions of Robotics and Automations, RA-3(3), 224-238.